UNIVERSITI PUTRA MALAYSIA

MINIMIZATIONOF TEST CASES AND FAULT DETECTION EFFECTIVENESS IMPROVEMENT THROUGH MODIFIED REDUCTION WITH SELECTIVE REDUNDANCY ALGORITHM

SHIMA NIKFAL

FSKTM 2007 20

MINIMIZATIONOF TEST CASES AND FAULT DETECTION EFFECTIVENESS IMPROVEMENT THROUGH MODIFIED REDUCTION WITH SELECTIVE REDUNDANCY ALGORITHM

By

SHIMA NIKFAL

Thesis Submitted to the School of Graduate Studies, Universiti Putra Malaysia, in Fulfilment of the Requirements for the Degree of Master of Science

December 2007

To

My Beloved Father and Mother,
My Brother and Sisters

Abstract of thesis presented to the Senate of Universiti Putra Malaysia in fulfilment of the requirement for the degree of Master of Science

MINIMIZATION OF TEST CASES AND FAULT DETECTION EFFECTIVENESS IMPROVEMENT THROUGH MODIFIED REDUCTION WITH SELECTIVE REDUNDANCY ALGORITHM

By

SHIMA NIKFAL

December 2007

Chairman: Associate Professor Abdul Azim Abd. Ghani, PhD

Faculty: Computer Science and Information Technology

In any software development lifecycle, testing is necessary to guarantee the quality of the end product. As software grows, the size of test suites grows too. Due to this grows, maintaining of test suites become more difficult. Therefore, test suite minimization techniques are required to control the test suite size. One way of doing this is by ensuring that the set of test suite includes the important test cases with all redundancies in test cases eliminated.

Most test suite minimization techniques remove redundant test cases with respect to a particular coverage criterion at a time. A potential drawback of these techniques is that they may result in loss of test suite coverage with respect to other coverage criteria, thus affecting the ability of reduced test suite in detecting faults.

To overcome this weakness, this research objective is to minimize the test suite by selectively including coverage redundancy while improving fault detection effectiveness. To achieve such goal, this research modifies and improves the Reduction with Selective Redundancy (RSR) algorithm.

In the modify algorithm, test cases would be selected according to the branch coverage if they covered different branch combination. Then the algorithm gathers all the test cases based on the definition occurrence and def-use pair if they cover same definition occurrence of one variable but they don't cover def-use pair of the same variable. Among these selected test cases, the algorithm identifies the redundant test cases based on definition occurrence, if they cover a similar combination of branch coverage except in one branch and also if the test cases cover a similar definition occurrence .

The results show the algorithm used in this research can reduce the test suite size as well as significantly improve the fault detection effectiveness. The fault detection loss of reduced suite size was significantly less than the amount of suite size reduction. Moreover, the results reveal that test suit minimization based on branch combination is effective in term of faults detection.

Abstrak tesis yang dikemukakan kepada Senat Universiti Putra Malaysia
sebagai memenuhi keperluan untuk ijazah Master Sains

PENGURANGAN KES UJIAN DAN PENAMBAHBAIKAN KEBERKESANAN PENGESANAN KECACATAN MELALUI PERUBAHAN ALGORITMA PENGECILAN DENGAN PEMILIHAN BERLEBIHAN

Oleh

SHIMA NIKFAL

Disember 2008

Pengerusi: Associate Profesor Abdul Azim Abd. Ghani, PhD

Fakulti: Sains Komputer dan Teknologi Maklumat

Dalam mana-mana kitaran pembangunan perisian, pengujian diperlukan untuk menjamin kualiti produk yang dihasilkan. Apabila berkembangnya perisian, saiz suit ujian perisian juga turut berkembang. Berdasarkan pengembangan ini, penyenggaraan terhadap suit ujian akan menjadi lebih sukar. Sehubungan dengan itu, satu teknik pengurangan suit ujian diperlukan untuk mengawal saiz suit ujian yang dihasilkan. Salah satu cara ialah dengan memastikan setiap suit ujian mengandungi kes pengujian yang terpenting sahaja dan mana yang berulang akan dihapuskan.

Kebanyakan teknik pengurangan menghapuskan kes pengujian yang berulang berdasarkan beberapa kriteria liputan yang tertentu. Kelemahan teknik ini ialah ia mungkin mengurangkan liputan suit ujian berdasarkan kriteria liputan tertentu, sekaligus mempengaruhi kebolehan suit ujian untuk mengesan kecacatan.

Untuk mengatasi kelemahan ini, objektif penyelidikan ini adalah untuk mengurangkan suit ujian dengan membuat pemilihan termasuk keberulangan liputan sementara menambahbaik keberkesanan pengesanan kecacatan. Untuk mencapai objektif tersebut, penyelidikan ini merubah dan menambahbaik teknik pengurangan suit ujian terkini yang dinamakan Pengecilan dengan Pemilihan Berlebihan.

Dalam algoritma yang diubah, kes-kes ujian akan dipilih berdasarkan kepada liputan cabang jika ianya meliputi kombinasi cabang yang berbeza. Kemudian algoritma tersebut mengumpul semua kes ujian berdasarkan kepada takrifan kejadian dan pasangan *def-use* jika ianya meliputi takrifan kejadian satu pembolehubah yang sama, tetapi ianya tidak meliputi pasangan *def-use* pembolehubah yang sama. Diantara kes ujian terpilih, algoritma tersebut mengenal pasti kes ujian berulang berdasarkan takrifan kejadian jika ianya meliputi kombinasi serupa liputan cabang kecuali dalam satu cabang dan juga jika kes ujian meliputi takrifan kejadian serupa.

Keputusan menunjukkan bahawa algoritma yang digunakan dalam penyelidikan ini dapat mengurangkan saiz suit ujian sementara menambahbaik keberkesanan pengesanan kecacatan. Kerugian pengesanan kecacatan untuk saiz suit yang dikurangkan adalah lebih kecil daripada amaun pengurangan saiz suit. Selain daripada itu, keputusan juga mendedahkan bahawa pengurangan suit ujian berdasarkan kepada kombinasi cabang adalah efektif dari segi pengesanan kecacatan.

ACKNOWLEDGEMENTS

First and foremost I would like to express my deep gratefulness to my parent for providing me the opportunity to continue my master’s program and financial support. And I’m grateful to my supervisor Associate Professor. Dr. Abdul Azim Abdul Ghani, for his kind assistance, critical advice, encouragement and suggestions during the study and preparation of this thesis. Moreover, I appreciate his encouragement to provide the opportunity to attend several conferences. I truly appreciate the time he devoted in advising me and showing me the proper directions to continue this research and for his openness, honesty and sincerity.

I would also like to express my gratitude to my co-supervisor Associate Professor. Hj.Mohd Hassan Selamat, to whom I’m grateful for his practical experience and knowledge that made an invaluable contribution to this thesis.

I also owe thanks to all of the people who were been willing to provide assistance and give advice. Last but not the least the deepest appreciation goes to my friends Fawzi Elfaidi, Farzaneh Abed and Mr. Mohamad Farid Bin Jaafar for their contentious support and encouragement. Another thank you goes to Ms. Fakariah Hani Mohd Ali for the translation of my abstract into Malay language.

I certify that an Examination Committee has met on 7 December 2007 to conduct the final examination of Shima Nikfal on her Master of Science thesis entitled " Minimization of Test Cases and Fault Detection Effectiveness Improvement through modified reduction with selective redundancy Algorithem " in accordance with Universiti Pertanian Malaysia (Higher Degree) Act 1980 and Universiti Pertanian Malaysia (Higher Degree) Regulations 1981. The Committee recommends that the candidate be awarded the degree of Master of Science.

Members of the Examination Committee are as follows:

Rahmita Wirza O.K Rahmat, PhD
Lecturer
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Chairman)

Masrah Azrifah Azmi Murad, PhD
Lecturer
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Internal Examiner)

Rusli Abdullah, PhD
Lecturer
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Internal Examiner)

Shamsul Sahibuddin, PhD
Associate Professor
Faculty of Computer Science and Information Technology
University of Technology Malaysia
(External Examiner)

HASANAH MOHD GHAZALI, PhD
Professor and Deputy Dean
School of Graduate Studies
Universiti Putra Malaysia

Date: 29 Januari 2008

This thesis was submitted to the Senate of Universiti Putra Malaysia and has been accepted as fulfilment of the requirement for the degree of Master of Science. Members of the Supervisory Committee were as follows:

Abdul Azim Abdul. Ghani, PhD
Associate Professor
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Chairman)

Mohd Hassan Selamat, PhD
Lecturer
Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
(Member)

AINI IDERIS, PhD
Professor and Dean
School of Graduate Studies
Universiti Putra Malaysia

Date: 21 February 2008

DECLARATION

I hereby declare that the thesis is based on my original work except for quotations and citations which have been duly acknowledged. I also declare that it has not been previously or concurrently submitted for any other degree at UPM or other institutions.

SHIMA NIKFAL

Date:

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

LIST OF ABBREVIATIONS

ATAC	Automatic Test Analysis for C
ATACMIN	Automatic Test Analysis for C Minimization
BVA	Boundary Value Analysis
C-use	Computation Use
DC	Decision Coverage
Def	Definition
EP	Equivalence Partitioning
FPC	Full Predicate Coverage
HGS	Harrold Gupta Soffa
ITS	Initial Test Suite
MC/DC	Modified Condition/ Decision Coverage
MRSR	Modified Reduction with Selective Redundancy
PC	Primary Criterion
P-use	Predication Use
RS	Reduce Suite
RSR	Reduction with Selective Redundancy
PSSC	Probabilistic Statement Sensitivity Coverage
RTC	Redundant Test Case
SC	Secondary Criterion
T	Test case
TC	Tertiary Criterion
USC	Uncovered Secondary Criterion
W.R.T	With Respect To

CHAPTER 1

INTRODUCTION

1.1 Background

Software development lifecycle involves a series of production activities. These activities create software requirement, generate the software specification and implement the software. During these activities, some errors may occur. Therefore, software should be comprehensively tested to remove errors and to ensure that the software meets its specification. Since, developed software go through maintenance, software may be changed over time. Due to these changes, testing and retesting of software occur continuously during the software development lifecycle.

Software testing is a process or a series of processes for analyzing the developed software to make sure that the actual behavior of the software correctly followed its specification. The essence of software testing is to execute the software with a particular set of input and observing the actual software output then comparing the gained output with the expected output. This particular set of program input along with the corresponding expected output is called a test case, while a group of test cases is called as a test suite or a test set.

Each existing test case in a test suite covers (exercises) some particular software testing requirements. A software testing requirement can be either Black box (specification based) or White box (code-based). White box requirements contain Statements, Decisions, Definition-use pair coverage. Black box requirements contain the coverage of special input values and output values which are generated from the specification. A test case is generated normally to cover a particular requirement or set of requirements, while covering more requirement hints that most of the software has been tested.

Software grows and evolves, along with growing of the test suites. More test cases are needed due to these progressive changes. Over time, several test cases in a test suite can cover specific requirements which may be covered by other test cases in the test suite. Therefore, the mentioned test cases become redundant considering a particular coverage criterion.

For instance, a test case is redundant with respect to statement coverage if it covers a particular set of statements which already have been covered by other test cases in the test suite. However, the same test case may actually not be redundant with respect to another coverage criterion, such as, definition-use pair coverage. Thus, it is important to consider that a test case redundancy is a related property to some specific set of requirements.

Since test suites size increases and it can be so large besides, test suites can often be used for retesting the software, the testing process will be so expensive. Due to time and resource limitations for testing the software it is essential to decrease the suites size. A reduction in the size of the test suite decrease both the overhead of maintaining the test suite and also the number of test cases that must be rerun after changes are made to the software. Test suite minimization is one general technique that has been proposed to address the problem of extremely large test suites.

Minimization techniques attempt to remove test cases from test suite however, removing the test cases can make a minimized suite weaker than un-minimized suite on detecting fault in software. Therefore a test suite minimization problem is an instance of a more general set-cover problem.

Set-cover problem is to find a minimally-sized subset of *S*, while a collection *S* of sets covers a particular group of entities, which provides the same amount of entity coverage as the original set *S*. The set-cover problem has been shown to be NP-Complete (Harrold *et al*., 1993), and therefore in general, no polynomial-time algorithm exists to optimally solve the minimization problem. Nevertheless, there has been some research works (Black *et al.*, 2004; Horgan and London, 1992) in the area of computing optimally-minimized suites. Most of other research works in minimization has relied on heuristics for computing near-optimal solutions.

Jones and Harrold (2003) described two minimization heuristics which are designed specifically to be used in conjunction with the relatively strong modified condition/decision coverage criterion; one algorithm builds a minimized suite incrementally by identifying essential and redundant test cases, while the other algorithm is based on a prioritization technique that simply stops computing before all test cases in a suite have been prioritized.

Agrawal (1999) implied a framework for minimization of suites using the notions of mega blocks and global dominator graphs. An algorithm based on a greedy heuristic for reducing the size of a test suite (referred to henceforth as the HGS algorithm) was developed by Harrold *et al*., (1993).

Jeffrey and Gupta (2005) presented an algorithm, named Reduction wit Selective Redundancy (RSR), based on a greedy heuristic to reduce the size of suite with selective redundant test case retain to decrease the loss of fault detection effectiveness. There are two coverage criteria used in RSR algorithm which are Branch coverage and data flow coverage used to identify a redundant test case. The test case becomes redundant if it does not cover any new branch and at the same time does not cover any new Definition-use pair. Therefore, in RSR algorithm if all the branches are covered once, then different combinations of these branches can't be recognized.

1.2 Problem Statement

Test suite minimization problem is to find a minimal subset of the test cases in a test suite that exercises the same set of coverage requirements as the original suite. The key idea behind minimization techniques is to remove the test cases in a test suite that have become redundant with respect to the coverage of some particular set of program requirements. On the other hand, the purpose of test cases is to reveal faults in software.

By removing test cases from test suite, the minimized suites may be weaker than their non-minimized counterparts on detecting faults in software. Hence, fault detection effectiveness is intuitively a measure of the ability of a test suite to detect faults in software. Therefore, in this thesis, it is attempted to improve the fault detection capabilities of reduced suites without significant effect on suite size reduction.

Jeffrey and Gupta (2005) suggested that it is possible to achieve high suite size reduction with little loss in fault detection effectiveness by keeping certain test cases that are redundant with respect to the particular coverage criterion. The RSR algorithm selects test cases considering the most uncovered coverage criterion. Here, if some test cases cover the maximum number of uncovered coverage criterion then among those test cases, one arbitrary test case is selected. The arbitrarily selection affects the ability of fault detection of the test suite reduction.

To achieve this goal, test suite sizes will be small — but not necessarily minimal — with respect to minimization criteria. Since, removing redundant tests from a suite based on one criterion will throw away some important tests that are not redundant according to other criteria in most of the cases. The following is suggested: test suite reduction with the goal of achieving high suite size reduction with little loss in fault detection effectiveness, in general, should incorporate some notion of keeping certain redundant test cases with respect to the particular set of program requirements by which minimization is carried out.

1.3 Research Objectives

The main objective of this research is to improve a RSR minimization algorithm to decreasing fault detection effectiveness loss. Details objectives are as follows:

- To propose Modified RSR (MRSR) algorithm to reduce the test suite size.
- To improve RSR algorithms to increase fault detection effectiveness.

1.4 Research Scope

This research is scoped according to the following delimitations:

- The program under test has been written in C++ language. The program is free from syntax errors.
- Very simple code has been considered that contains one function with not more than 5 IF statement and each IF statement has one condition not more.
- Loop, pointer, string and array are not considered.

1.5 Research Contributions

Many test suite minimization algorithms have been proposed to minimize the number of test cases existing in a test suite without effecting on the fault detection effectiveness. The main contribution of this research is:

- A test suite minimization algorithm is developed to produce a reduced suite with a high chance of fault detection, by keeping some redundant test cases

1.6 Thesis Organization

This thesis is outlined in 7 chapters. This chapter provides background information about test suite minimization, and explains the problem statement. The objective and contribution of this research is also included in this chapter.

Chapter 2 consists of the reviewed literature of the related works. Furthermore the related works to test suite minimization and fault detection effectiveness have been discussed later. Chapter 3 contains a general description of research methodology. It explains the test suite generation, proposed test suite minimization method, experimental design and evaluation methods. Chapter 4 is a detailed description of test suite generation, developing the scanner and developing the test cases and test cases coverage computing.

In chapter 5 the improvement of test suite minimization algorithm has been discussed and in it is illustrated that how this method works in later part of the chapter as a case study. In chapter 6 the experimental design and the performance of MRSR algorithm is evaluated, and in the last section comparative analysis has been discussed. Chapter 7 shows the conclusion that summarizes the most important aspects of research. This chapter ends with suggested future works.